第1章　最適化問題

1.1　最適化問題とは何か

例えば，あなたが自宅から会社もしくは学校に移動する場合を考えて下さい．多くの人は，最短の時間で会社 (学校) にたどり着ける交通機関の組合せやルートの選択を行っていることと思います（図 1.1）． このような最短経路を求める問題は，最適化問題の有名な問題として知られています．

最適化問題 (optimization problem) とは，

”与えられた制約条件の下で，ある目的関数を最大 (小) にする解を求める”

と定義することが出来ます．最適化問題は，最短経路問題の様な身近な問題に限らず，工学，経済学を始めとする様々な分野で研究が行われています．また，最適化問題と非常に関連のある分野としてはオペレーションズ・リサーチ (OR,Operations Reseaech) やゲーム理論等があります．

1.2　最適化問題の解法

一般に，ある最適化問題を解くにはその問題の背景に十分な注意を払って，制約条件や目的関数の評価基準を定め，数式もしくはグラフ等のモデルによって最適化問題を単純化します．そして，単純化された問題に対してアルゴリズム (問題を解く手順の記述) を適用し，主にコンピュータ上でシミュレーションを行うことによって最適化問題の解を得ることが出来ます．

最適化問題に対するアルゴリズムには，オペレーションズ・リサーチや数理計画法の他に，人工知能 (AI)，エキスパートシステム，システム理論，ファジイ集合，ニューラルネットワーク，遺伝的アルゴリズム，等いろいろなパラダイムが存在しています．また，最適化問題にも連続値を扱う問題と離散値 (整数, 集合, グラフなど) を扱う問題があります．後者の離散値 (整数のみ) を扱う問題は特に組合せ最適化問題と呼ばれます．組合せ最適化問題には一般の最適化問題の解法がほとんど適用できないため，数え上げの手法に基づく独自の解法が考案されており，活発な研究がなされています．

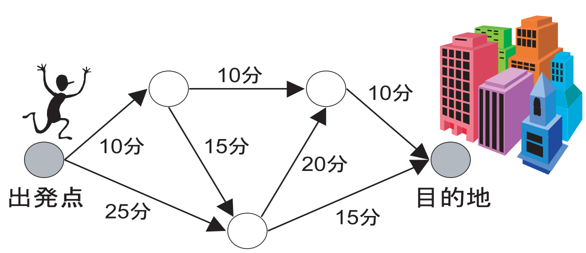

図 1.1: 経路の最適化

表 1.1: 各食品 100g 中の栄養含有量と各栄養素の最低必要量

	食品 1	食品 2	最低必要量
栄養素 A	8	1	20
栄養素 B	2	2	12
栄養素 C	2	5	20

1.3 最適化問題の例

簡単な線形計画問題を例にとって，最適化問題とは具体的にどんなものなのかを考えてみましょう.

問題

人間が生きていくために必要な栄養素をできるだけ安価に摂取するという問題を考えます．いま，食品に含まれる各栄養素の量と，生きていくために最低限必要な栄養素の量は表 1.1 のようになっているとします.
食品 1 および 2 の 100g あたりの価格を 300 円，100 円として，必要とされる栄養素をできるだけ安価に摂取するためには，食品 1 および 2 を何 g ずつ食べればよいでしょうか？

1.3.1 目的関数の定式化

食品 1 を x_1g，食品 2 を x_2g 購入すると，かかる費用は

$$300x_1 + 100x_2 \quad （円） \tag{1.1}$$

と表せます．この問題ではいかに安く必要な栄養素を摂取するかが目的ですから，この式を最小にする x_1 と x_2 を求めればよいと言うことになります.

$$f = 300x_1 + 100x_2 \longrightarrow min \tag{1.2}$$

この関数を**目的関数**と呼びます．また x_1，x_2 を**設計変数**と呼びます．このままでは，問題が解きにくいため，式を変形し

$$F = 3x_1 + 1x_2 \quad (F = f/100) \tag{1.3}$$

を最小化することを目的とします．これをグラフ化したものが図 4.11 です．F を最小にすることは x_2 切片にあたる F を最小にすることと同義です．つまり，この目的関数を $(x_1, x_2) = (0, 0)$ により近づけることが最小値を見つける近道となるのです．したがって，目的関数を図 4.11 の矢印の方向に動かせばよいのです.

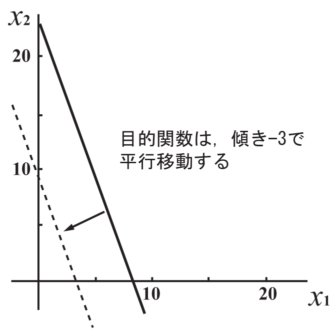

図 1.2: 目的関数

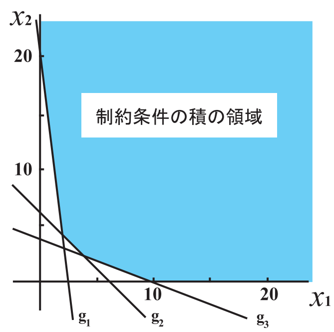

図 1.3: 制約条件

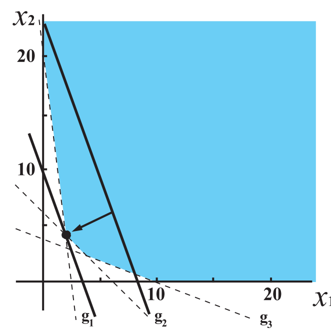

図 1.4: 最適解の探索

1.3.2 制約条件の定式化

この問題には，各栄養素を必要最低量以上摂取しなければならないと言う条件があります．このような満たすべき条件のことを**制約条件**と呼びます．

表 1.1 より，栄養素 A は食品 1 から 8g，食品 2 から 1g 摂取することができます．また，栄養素 A は，1 日に最低 20g 必要とされているので

$$g_1 = 8x_1 + x_2 \geq 20 \tag{1.4}$$

と表すことができます．同様に，栄養素 B，C についても定式化すると制約条件は次の 3 式で表せます．また，これを，グラフ化したものが図 1.3 です．

$$\begin{array}{ll} \text{栄養素 } A： & g_1 = 8x_1 + x_2 \geq 20 \\ \text{栄養素 } B： & g_2 = 2x_1 + 2x_2 \geq 12 \\ \text{栄養素 } C： & g_3 = 2x_1 + 5x_2 \geq 20 \end{array} \tag{1.5}$$

1.3.3 解の探索

図 1.3 の網掛け部分が，制約条件を満たしている部分です．したがって，この範囲の中で，目的関数が最小になるものが最適解であるといえます．図 1.4 のように，目的関数を動かします．

この場合，最適解は図 1.4 より，$x_1 = 2(\times 100\mathrm{g})$　$x_2 = 4(\times 100\mathrm{g})$ となり，このときの目的関数の値は $f = 1000$（円）となります．

最適解を見つけるためのアルゴリズムが多数存在します．その中でも代表的なものについて●章で紹介します．

第2章　最適化問題の分類

最適化問題は，その数学的な性質によっていくつかの問題に分類することができます．この章では，それらのうち，線形問題，非線形問題，離散問題という基本的な問題について紹介します．

2.1　線形・非線形計画問題

2.1.1　線形計画問題

線形計画問題（Linear programming problem, LP）とは，モデルの目的関数と制約条件が，すべて線形である（変数に関する１次式で表現できる）ような問題を指します．前章で取り上げた例題も線形問題です．

線形問題は，一般的に次のように表すことができます．

$$\text{目的関数} \quad f = c_1x_1 + c_2x_2 + \cdots + c_nx_n \longrightarrow \max\ (\mathit{or}\ \ \min) \tag{2.1}$$

$$\begin{aligned} \text{制約条件} \quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n && \le b_1 \\ & a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n && \le b_2 \\ & \vdots && \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n && \le b_m \end{aligned} \tag{2.2}$$

これを行列・ベクトル表現を用いて簡潔に表すことができます．

A, b, c, x をそれぞれ

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_3 \end{pmatrix}, \quad c = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_3 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_3 \end{pmatrix} \tag{2.3}$$

と置くと線形問題は，

制約条件 $Ax \le b$ のもとで，目的関数 $f(x) = c^\top x$ を最小化するような変数ベクトル $x = \{x_1, x_2 \cdots, x_n\}^\top$ を見いだす問題

であるといえます．

表 2.1: 代表的な離散的計画問題

最短経路問題	A 点から B 点までの最短の経路を見いだす問題（図 1.1）
巡回セールスマン問題 (TSP)	複数の都市を巡回するための最短のルートを見いだす問題
ナップザック問題	総重量 x 以下という条件の下で，より多くのものをナップサックに詰めるためにはどうすればよいかという問題
スケジューリング問題	m 種の機械を用いて n 個の作業を実行するとき，各機械上での作業順序を決定する問題
工場設置問題	工場をどこに設置するのが建設費と製品の輸送コストの観点から最適かを問う問題

線形計画問題のアルゴリズム

線形計画問題に対するアルゴリズムでもっとも一般的なものは，**シンプレックス法**（simplex method）です．シンプレックス法は，一般の LP 問題に対してなるべく効率よく，解の現れる端点を探し出そうとするアルゴリズムです．シンプレックス法では，解が存在するか否かをはじめに判定する過程，および解が有界か否かを判定する過程も含むように工夫されており，収束性も保証されていることから，極めて優れた一般性のあるアルゴリズムであるといえます．ただし，設計変数の数が多い大規模線形計画問題には適さないという欠点もあります．

2.1.2 非線形計画問題

非線形計画問題（Nonliner programming problem）とは，目的関数または制約条件の少なくとも一方が非線形関数であるような最適化問題のことを指します．

非線形計画問題のアルゴリズム

非線形計画問題に対するアルゴリズムとしては，Newton 法や準 Newton 法が一般的です．これは目的関数の勾配ベクトルと，Hesse 行列を利用することで直線上の最小化を繰り返し，最適解を効率的に求めるアルゴリズムです．Newton 法は，解の近傍での収束速度は非常に速いのですが，諸基点を解の近傍にとらなければ収束性が保証されないという欠点があります．また，Hesse 行列の計算や線形代数方程式を解かなければならず，計算量が多いことも問題です．

2.2 離散問題

離散的計画問題（discrete programming problem）とは，目的関数や制約条件が離散集合である最適化問題のことを指します．離散的なものの例としては，整数の集合，グラフにおけるノードやアークの集合などが挙げられます．離散集合では，連続性，微分，積分などの概念を直接適用することができないという特徴があります．

ここで，離散的計画問題の代表例を表 2.1 いくつか挙げておきます．このように，離散的計画問題は工学のほとんどすべての分野に登場する非常に重要な問題です．しかしながら，離散問題には，連

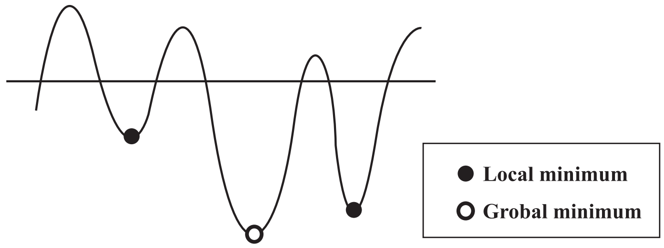

図 2.1: 大域的最小点と局所的最小点

続性や微分といった古典的な手法をとることができない難しさがあります．また，各分野で生じる問題は多様であるため，それらに共通する構造を見抜き一般化することも困難であるといえます．このような数学的なモデル化が困難な問題を解決するのに GA（遺伝的アルゴリズム）が有効であることが知られています．

2.3 その他

よく使う微係数

目的関数 $f(x)$ が微分可能であるとき，最適化のアルゴリズムで目的関数の 1 次または 2 次の微係数を用いることがあります．そこでは

$$\nabla f \triangle \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots, \frac{\partial f}{\partial x_n}\right)^\top \quad (n\text{ ベクトル}) \tag{2.4}$$

という**勾配ベクトル** (gradient vector)，および

$$H \triangle \nabla^2 f = \left(\frac{\partial^2 f}{\partial x_i \partial x_j}\right) \quad (n \times m\text{ 行列}) \tag{2.5}$$

という **Hesse 行列** (Hessian) がよく登場します．なお，ベクトルや行列の転置は，右肩に添字 $\top$ をつけて表します．

大域的最小点と局所的最小点

最適化問題における難問の一つに，大域的最小点と局所的最小解の問題があります．制約条件 L を満たす設計変数 $x_0, (x_0 \in L)$ について

$$f(x_0) \leq f(x) \qquad (\forall x \in L) \tag{2.6}$$

であるとき，x_0 を**大域的最小点（global minimum）**といいます．それに対して，$x_0 \in L$ について，x_0 の近くにあるすべての $x \in L$ に対してのみ

$$f(x_0) \leq f(x) \tag{2.7}$$

であるとき，x_0 を**局所的最小点（local minimum）**といいます．

一般に局所的最小点は互いに離れていくつも存在します．一つの局所的最小点を求めることは比較的簡単ですが，すべての局所的最小点を求めて，その中から大域的最小点，つまり真の最適解を求めることは非常に困難です．

第3章　最適化のアルゴリズム

3.1　二分法のアルゴリズム

連続関数 $f(x)$ の値が，2 点 x_1, x_2 で異なる符号を持つならば，その間に少なくとも 1 つの零点がある．そこで，

$$f(x_1) \cdot f(x_2) < 0$$

という条件を満たす 1 組の x_1, x_2 から出発し，次にその中点

$$x_m = \frac{1}{2}(x_1 + x_2)$$

での $f(x)$ の符号を調べる．その結果，$f(x_1) \cdot f(x_m) > 0$ であれば x_1 を x_m で置き換え，そうでなければ x_2 を x_m で置き換える．これを繰り返し，あらかじめ定めた ϵ に対して

$$|x_1 - x_2| < \epsilon$$

が成り立てばその時点で打ち切る．

3.2　最急降下法のアルゴリズム

非線形計画問題のうち，制約条件がない問題に対して有効な解法が最急降下法である．
最急降下法のアルゴリズムを説明する．

1. **最急降下法**
   初期点 x_k を定め，その点における微分係数（勾配）$f'(x_k)$ を求める．勾配は，その点において目的関数 $f(x)$ の値が最も大きく増加する方向であるので，関数 $f(x)$ の値を減少させるため，勾配と逆の方向 $d_k = -f'(x_k)$ に進めばよい．

$$x_{k+1} = x_k + \alpha_k d_k$$

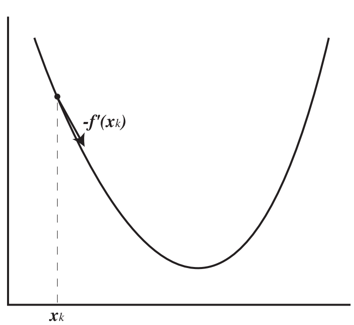

図 3.1: 最急降下法

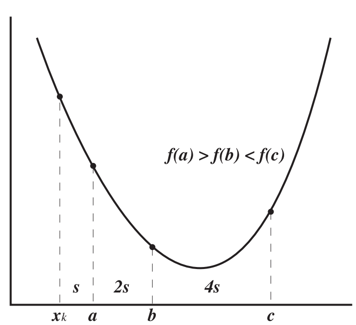

図 3.2: 直線探索

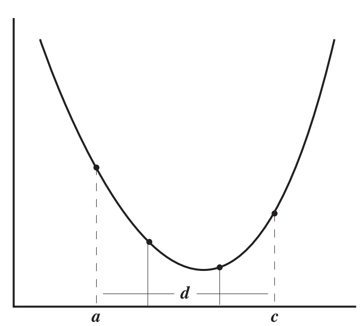

図 3.3: 黄金分割法

を次の点とすればよい．ここで α_k はステップ幅という．

2. **直線探索**
ステップ幅を具体的に求めるため，直線探索を行う．初期点 x_k より，上で求めた探索方向に沿って，最小点を通り越すまでだんだん増加して進んでいく．ここでは微少な距離 s を設定し，$s, 2s, 4s$ のように進める．各々の点での関数値 $f(x_i), n$: 自然数 を計算し，$f(x_i) < f(x_{i+1})$ となった時点で $x_{i-1} = x_1, x_{i+1} = x_2$ として黄金分割法に進む．

3. **黄金分割法**
黄金比 $\frac{1+\sqrt{5}}{2}$ を τ とする．x_1 と x_2 の距離を d とし，新たな点

$$x_3 = x_1 + \frac{\tau - 1}{\tau} d, x_4 = x_1 + \frac{1}{\tau} d$$

を求める．関数値 $f(x_3), f(x_4)$ を求め，$f(x_3) > f(x_4)$ ならば $x_1 = x_3$ とし，$f(x_3) < f(x_4)$ ならば $x_2 = x_4$ とする．これを繰り返し，あらかじめ定めた ϵ に対して $d < \epsilon$ が成り立てばその時点で打ち切る．